use the direct method instead

The above figure represents a simple example of the kind of problem queuing theory can be applied to. Some examples of such problems are:

- People choosing a line at a store.
- Packets arriving to a router/switch.
• Calls to a customer service center.
• People booking flights.
• In general, any instance of Buffering/Waiting in queue until service happens is applicable.

1 Some Definitions

• Service Mechanism: How customers get served.

• Service Discipline: Who gets served. eg.
  – FIFO – first in first out
  – LIFO – last in first out
  – Random
  – Prioritized Queues
  – M/M/1 queue - the classic example.

• Service Time: Time to complete 1 job in queue.

• Inter-arrival time: time between 2 arrivals.

• Queueing System Notation a/b/c
  a: Distribution of inter-arrival times. b: Distribution of service times. c: of servers.
  eg. $\mu/\mu/1$ or $c/\mu/1$
  $\mu$ – Markovian/Exponential/Memoryless
  $D$ – Deterministic/Degenerate/Constant
  $Ea$ – Erlang distance with parameter $k$.
  $G$ – general

Memory less implies that past state is not taken into account.
For example, if you are in line waiting for a bus which on average takes about 20 minutes to arrive.
Say 15 minutes have passed. A memoryless model would imply that you still have to wait around 20 minutes for the bus!

• Queue length can be finite or infinite. Analysis is easier for the infinite case.

• State – num. customers in system.
• Queue Length – num. customers waiting = state - (num. being served)
• \(N(E)\) – num. customers at time \(t\)
• \(P_n(t)\) – Probability that there are \(n\) customers at time \(t\) (given information about the time).
• \(S\) = num. parallel servers
• \(\lambda_n\) – mean arrival time when there are \(n\) in system.
• \(\mu_n\) – mean service rate when there are \(n\) in system
• \(C\) – combined rate of completion.
• If \(\lambda_n\) is constant it is called just \(\lambda\) when the service rate is constant per server.
• \(e\) – utilization rate - factor. Generally, if \(e < 1\), then we can get stability in the system.
• \(L\) – expected num. in system. \(L\) is equal to \(\sum_{n=0}^{\infty} nP_n\)
  where \(P_n\) – the probability that \(n\) are in the system.
• \(L_q\) – expected num. in queue \(\sum_{n=0}^{\infty} (n - s)P_n\)
• \(W\) – expected waiting time
• \(W_q\) – expected time in queue

2 Little’s Law

\[
L = \lambda W \\
L_q = \lambda W_q
\]

If mean service is constant \(1/\mu\) for \(a \geq 1\) then \(W = W_q + 1/\mu\).
This means that if we have one of \(L,W,L_q,W_q\) we can calculate them all!

3 Common Assumptions

Arrivals follow an exponential distribution:

\[
f_T(t) = \begin{cases} \alpha e^{-\alpha t} & t \geq 0 \\ 0 & t \leq 0 \end{cases}
\]
\[ P(T \leq t) = 1 - e^{-\alpha t} \]
\[ P(T \geq t) = e^{-\alpha t} \]
\[ E(T) = \int_{0}^{\infty} t f_T(t) dt = 1/\alpha \]
\[ var(t) = ET^2 - (ET)^2 = 1/\alpha^2 \]

Is this a reasonable model? Yes, for many systems such as phone calls and large shops.

4 Nice Properties of exponential

\[ P(T > t + \triangle t | T > t) = \frac{P(T > \triangle t, T > t + \triangle t)}{P(T > \triangle t)} \]
\[ = \frac{P(T > t + \triangle t)}{P(T > \triangle t)} \]
\[ = \frac{e^{-\alpha(t+\triangle t)}}{e^{-\alpha t}} \]
\[ = e^{-\alpha t} \]
\[ = P(T > t) \]

This implies that it does not matter how long we have been waiting, hence state information is very simple. We only need to keep track of the number in the system.

4.1 Another Interesting Property

The mch of a set of exponentials is exponential. Let \( u = \{\text{mch}\}T_1, \ldots T_n \)
\[ P(U > t) = P(T_1 > T, T_2 > T, T_3 > T, \ldots T_n > T) \]
\[ = P(T_1 > t)P(T_2 > t)P(T_3 > t) \ldots P(T_n > t) \]
\[ = e^{-\alpha_1 t} e^{-\alpha_2 t} \ldots e^{-\alpha_n t} \]
\[ = e^{-\sum_{i=1}^{n} \alpha_i t} \]

The service times are exponential and the queue length is Poisson.
\[ P(X(E) = n) = \frac{(\alpha t)^n e^{-\alpha t}}{n!} \]
4.2 Properties of a Poisson Process

\[ P(X(t) = 0) = e^{-\alpha t} \]  
\[ E(x(t)) = \alpha t \]

\( \alpha = \text{arrivals per unit time} \)

It has similar properties to that of the exponential process.

5 Example

Say we have one queue and are interested in the number of jobs waiting. Assume all rates are independent. \( P_n \) = probability that there are \( n \) in the system. Eventually the system will go to a steady state.

In that state we have the following balanced equations:

\[ \lambda_1 P_1 = \lambda_0 P_0 \]
\[ \lambda_0 P_0 + \mu_2 P_2 = (\lambda_1 + \mu_1) P_1 \]
\[ \lambda_1 P_1 + \mu_3 P_3 = (\lambda_2 + \mu_2) P_2 \]

Leads to \( P_n = C_n P_0 \)

Where:

\[ C_n = \frac{\lambda_{n-1} + \lambda_{n-2} + \ldots + \lambda_0}{\mu_n \mu_{n-1} \ldots \mu_0} P_0 = \frac{1}{\sum_{n=0}^{\infty} C_n} \]

More details specific to each type of system is available in the text.

6 Kevin’s Research

Kevin’s research looks at problems in network scheduling. It is applicable to switches, optical storage, and Storage Area Networks. It deals with resource allocation, parallel queue networks, forwarding queues, managing network of queues and service models at different prices.

In any smart queuing system there is a tradeoff between complexity and performance. The challenge is to produce an algorithm which smoothly manages this tradeoff.